A Low Complexity Algorithm and Architecture for Systematic Encoding of Hermitian Codes

Rachit Agarwal*†, Ralf Koetter‡ and Emanuel M. Popovici*§
* Department of Microelectronic Engineering, University College Cork, Cork, Ireland
† Microelectronics Application Integration Group, Tyndall National Institute, Cork, Ireland
‡ Institute for Communications Engineering, Technische Universitaet, Muenchen, Germany
§ Claude Shannon Institute for Discrete Mathematics, Coding and Cryptography, Ireland
Email: rachit.agarwal@ue.ucc.ie, ralf.koetter@tum.de, e.popovici@ucc.ie

***Abstract*— We present an algorithm for systematic encoding of Hermitian codes. For a Hermitian code defined over $GF(q^2)$, the proposed algorithm achieves a run time complexity of $O(q^2)$ and is suitable for VLSI implementation. The encoder architecture uses as main blocks q varying-rate Reed-Solomon encoders and achieves a space complexity of $O(q^2)$ in terms of finite field multipliers and memory elements.**

I. Introduction

Algebraic-Geometric (AG) codes [1] offer desirable properties such as large code lengths over small finite fields, the potential to find a large selection of codes and good error-correction at high code rates [2]. In recent years, an important class of one-point AG codes, called Hermitian codes, has been frequently discussed [3]-[6].

For a Hermitian code defined over $GF(q^2)$, a brute-force way to design an encoder is to multiply the information vector by a generator matrix. The space complexity of a serial-in serial-out architecture for this systematic encoder is $O(q^5)$ in terms of finite field multipliers and $O(q^3)$ in terms of memory elements. The encoder requires $2n$ clock cycles to generate a codeword of length n, thus, the latency is n.

By considering a Hermitian code as a superposition of several generalized Reed Solomon (RS) codes, an encoding scheme is introduced in [4]. In [5], an encoding algorithm by forming a bivariate information polynomial and evaluating this polynomial at every finite rational point on the Hermitian curve is proposed. However, both such schemes are nonsystematic and involve the evaluation of bivariate polynomials at n finite rational points, thus, they may not have efficient hardware architecture for implementations.

A computationally efficient approach for systematic encoding was proposed in [7]. A serial-in serial-out architecture for this approach was proposed in [8]. This architecture requires n clock cycles to encode a codeword of length n. The space complexity for this architecture is $O(q^3)$, both in terms of finite field multipliers and memory elements.

In this paper, we present an algorithm for systematic encoding and syndrome computation of Hermitian codes. We give an outline for the encoder architecture, which uses q varying-rate RS encoders as main blocks and requires $n^{2/3}$ clock cycles for encoding a codeword of length n. The space complexity of the architecture is $O(q^2)$ in terms of both, memory elements and finite field multipliers.

II. Hermitian Codes and Syndrome Computation

We consider codes from a Hermitian curve

$$\chi : x^{q+1} = y^q + y$$

over a finite field $\mathbb{F}_{q^2}$. The space $L(mP_\infty)$ consists of all functions on χ that have a pole of multiplicity at most m only at the unique point at infinity. For $L(mP_\infty)$, we choose a basis

$$L(mP_\infty) = \langle x^a y^b : aq + b(q+1) \leq m,\ 0 \leq a,\ 0 \leq b < q \rangle$$

Let y_0 be an element of $\mathbb{F}_{q^2}$ such that $y_0 + {y_0}^q = 1$. The affine rational points on χ are of the form

$$P_{\alpha,\beta} = (\alpha, \alpha^{q+1}(y_0 + \beta) + \delta(\alpha)\beta),$$

where δ is the Kronecker-delta and α and β represent arbitrary elements in $\mathbb{F}_{q^2}$ and $\mathbb{F}_q$ respectively.

Let ϵ be a primitive element in $\mathbb{F}_{q^2}$ and let γ be a primitive element in $\mathbb{F}_q$. We label the positions in a codeword by the corresponding elements $\alpha = \epsilon^i$, $\beta = \gamma^j$ and we thus naturally consider a codeword as a $q \times q^2$ matrix $\mathbf{c}$. Occasionally we will index elements in this array by elements of the fields $\mathbb{F}_{q^2}$ and $\mathbb{F}_q$, otherwise we index starting with 0.

A Hermitian code $C(m)$ is defined as

$$\{\mathbf{c} \in \mathbb{F}_{q^2}^{q^3} : \sum_{\alpha \in \mathbb{F}_{q^2}} \sum_{\beta \in \mathbb{F}_q} \mathbf{c}_{\beta,\alpha} f(P_{\alpha,\beta}) = 0,\ \forall f \in L(mP_\infty)\}$$

For an in-depth treatment of AG codes we refer to [9]. Throughout this paper we consider m and thus the Hermitian code as being fixed.

Given a $q \times q^2$ matrix $\mathbf{r}$ we can check if $\mathbf{r}$ is a codematrix in a Hermitian code by computing the syndromes

$$S_{a,b}(\mathbf{r}) = \sum_{\alpha \in \mathbb{F}_{q^2}} \sum_{\beta \in \mathbb{F}_q} \mathbf{r}_{\beta,\alpha} (x(P_{\alpha,\beta}))^a (y(P_{\alpha,\beta}))^b$$

$\mathbf{r}$ is a code-matrix iff $S_{a,b}(\mathbf{r})$ is zero for all $x^a y^b \in L(mP_\infty)$. Substituting the explicit form of the points we get

$$S_{a,b}(\mathbf{r}) = \sum_{\alpha \in \mathbb{F}_{q^2}} \sum_{\beta \in \mathbb{F}_q} \alpha^a (\alpha^{(q+1)}(y_0 + \beta) + \delta(\alpha)\beta)^b \mathbf{r}_{\beta,\alpha}$$

These equations can further be developed to give specific forms as shown in (1) and (2). From the structure of (2), it comes naturally to define a matrix as in (3) to convert the expression into a matrix multiplication. Similarly we define a matrix A' as

$$\mathbf{A}' = \begin{pmatrix} 1 & (\gamma^0)^0 & (\gamma^1)^0 & \dots & (\gamma^{q-2})^0 \\ 0 & (\gamma^0)^1 & (\gamma^1)^1 & \dots & (\gamma^{q-2})^1 \\ 0 & (\gamma^0)^2 & (\gamma^1)^2 & \dots & (\gamma^{q-2})^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & (\gamma^0)^{q-1} & (\gamma^1)^{q-1} & \dots & (\gamma^{q-2})^{q-1} \end{pmatrix}$$

For later use we give here the following Lemma.

A. Lemma 1

The $l \times l$ submatrices of A consisting of the elements indexed by i, j and that of A' consisting of the elements indexed by j, i, $i = q-l,\ q-l+1, \dots, q-1,\ j = 0,\ 1, \dots, l-1$ are non-singular.

Proof: This lemma follows in both cases from the properties of Vandermonde matrices. ■

It will be convenient to define an array $\overline{A}$ of matrices of type A and A'.

$$\overline{A} = (A_0, A_1, \dots, A_{q^2-1}),\ A_i = \begin{cases} A' & \text{i = } q^2\text{-1} \\ A & \text{otherwise} \end{cases}$$

Given a $q \times q^2$ array $\mathbf{r}$ with columns $\mathbf{r}_j$, we define a $q \times q^2$ matrix $\tilde{\mathbf{r}}$ with columns $\tilde{\mathbf{r}}_j$ as

$$\tilde{\mathbf{r}}_j = A_j \mathbf{r}_j$$

One of the main ingredients in both the syndrome calculation and a systematic encoding is the use of techniques for cyclic codes which are extended by one extra position. Let $\hat{a}(b) = \lfloor (m - (q-1-b)(q+1))/q \rfloor = max(a : x^a y^b \in L(mP_\infty))$.

B. Definition 1

Let an ordered set $\Re = \{\xi_0, \xi_1, \dots\}$ of elements from $\mathbb{F}_q$ be given. We define the code $EC(\Re, q)$ as

$$\{c \in \mathbb{F}_q^q : \sum_{i=0}^{q-2} c_i \xi^i = 0,\ \forall \xi \in \Re \backslash \{\xi_0\},\ c_{q-1} + \sum_{i=0}^{q-2} c_i {\xi_0}^i = 0\}$$

For the natural indexing of elements in $\mathbb{F}_q$ and $\mathbb{F}_{q^2}$ induced by γ and ϵ we have the following Lemma.

C. Lemma 2

Let a $q \times q^2$ matrix $\mathbf{r}$ be given. The matrix $\mathbf{r}$ is a code matrix in the Hermitian code $C(m)$ iff the ith row of $\tilde{\mathbf{r}}$ is a codeword in $EC((\epsilon^{0+i(q+1)}, \epsilon^{1+i(q+1)}, \dots, \epsilon^{\hat{a}(i)+i(q+1)}), q^2)$.

Proof: The proof follows immediately from the syndrome definition. ■

Codes of type $EC((\epsilon^{i(q+1)}, \epsilon^{1+i(q+1)}, \dots, \epsilon^{\hat{a}(i)+i(q+1)}), q^2)$ will play a central roll in the sequel. We define codes E_i as

$$E_i = EC((\epsilon^{0+i(q+1)}, \epsilon^{1+i(q+1)}, \dots, \epsilon^{\hat{a}(i)+i(q+1)}), q^2)$$

From Lemma 2 we can derive an efficient way to compute the syndrome for a Hermitian code. Given a received matrix $\mathbf{r}$ we obtain a matrix $\tilde{\mathbf{r}}$ with columns $\tilde{\mathbf{r}}_j = A_j \mathbf{r}_j$.

Given $\tilde{\mathbf{r}}$ we can easily solve the task of computing syndromes provided we can compute the corresponding syndromes for codes $E_i,\ i = 0, 1, \dots, q-1$.

III. Systematic Encoding

The idea behind the systematic encoding of Hermitian codes is to use the well known techniques for the systematic encoding of cyclic codes. Lemma 2 almost immediately gives a nonsystematic encoding procedure for Hermitian codes. To this end let $\tilde{\mathbf{r}}$ be a $q \times q^2$ matrix such that the jth row of $\tilde{\mathbf{r}}$ is a codeword in E_j. It follows from Lemma 2 that we can obtain a code-matrix for a Hermitian code by multiplying the columns of $\tilde{\mathbf{r}}$ with matrices A^{-1} and A'^{-1} respectively. We can obtain such a matrix $\tilde{\mathbf{r}}$ using eg. systematic encoding procedures for codes of type $E_j,\ j = 0, \dots, q-1$.

We will need A^{-1} and A'^{-1}.

A. Lemma 3

The matrices A and A' have inverses given in (4) and (5).

Proof: The inverse of A' is straight forward to verify. We only show the inverse of A. The rows of A^{-1} and the columns of A may be thought of as being indexed by elements of $\mathbb{F}_q$.

$$S_{a,b}(\mathbf{r}) = \begin{cases} \sum_{\alpha \in \mathbb{F}_{q^2} \backslash \{0\}} \sum_{\beta \in \mathbb{F}_q} \alpha^a \alpha^{b(q+1)} (y_0 + \beta)^b \mathbf{r}_{\beta,\alpha} & \text{a} \neq 0 \\ \sum_{\alpha \in \mathbb{F}_{q^2}} \sum_{\beta \in \mathbb{F}_q} (\alpha^{b(q+1)} (y_0 + \beta)^b + \delta(\alpha) \beta^b) \mathbf{r}_{\beta,\alpha} & \text{a = 0} \end{cases} \tag{1}$$

$$S_{a,b}(\mathbf{r}) = \begin{cases} \sum_{i=0}^{q^2-2} \epsilon^{i(a+b(q+1))} (y_0^b \mathbf{r}_{0,i} + \sum_{j=0}^{q-2} (y_0 + \gamma^j)^b \mathbf{r}_{j+1,i}) & \text{a} \neq 0 \\ \sum_{i=0}^{q^2-2} \epsilon^{ib(q+1)} (y_0^b \mathbf{r}_{0,i} + \sum_{j=0}^{q-2} (y_0 + \gamma^j)^b \mathbf{r}_{j+1,i}) + \sum_{j=0}^{q-2} \gamma^{jb} \mathbf{r}_{j+1,q^2-1} & \text{a = 0} \end{cases} \tag{2}$$

$$\mathbf{A} = \begin{pmatrix} (y_0+0)^0 & (y_0+\gamma^0)^0 & (y_0+\gamma^1)^0 & \dots & (y_0+\gamma^{q-2})^0 \\ (y_0+0)^1 & (y_0+\gamma^0)^1 & (y_0+\gamma^1)^1 & \dots & (y_0+\gamma^{q-2})^1 \\ (y_0+0)^2 & (y_0+\gamma^0)^2 & (y_0+\gamma^1)^2 & \dots & (y_0+\gamma^{q-2})^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (y_0+0)^{q-1} & (y_0+\gamma^0)^{q-1} & (y_0+\gamma^1)^{q-1} & \dots & (y_0+\gamma^{q-2})^{q-1} \end{pmatrix} \tag{3}$$

$$\mathbf{A^{-1}} = \begin{pmatrix} 1-(y_0+0)^{q-1} & (y_0+0)^{q-2} & \ldots & (y_0+0)^0 \\ 1-(y_0+1)^{q-1} & (y_0+1)^{q-2} & \ldots & (y_0+1)^0 \\ 1-(y_0+\gamma)^{q-1} & (y_0+\gamma)^{q-2} & \ldots & (y_0+\gamma)^0 \\ \vdots & \vdots & \ddots & \vdots \\ 1-(y_0+\gamma^{q-2})^{q-1} & (y_0+\gamma^{q-2})^{q-2} & \ldots & (y_0+\gamma^{q-2})^0 \end{pmatrix} \tag{4}$$

$$\mathbf{A'^{-1}} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & -1 \\ 0 & -(1)^{q-2} & -(1)^{q-3} & \ldots & -(1)^1 & -1 \\ 0 & -(\gamma)^{q-2} & -(\gamma)^{q-3} & \ldots & -(\gamma)^1 & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & -(\gamma^{q-2})^{q-2} & -(\gamma^{q-2})^{q-3} & \ldots & -(\gamma^{q-2})^1 & -1 \end{pmatrix} \tag{5}$$

Let C be the matrix obtained as $C = A^{-1}A$. The entry $C_{i,j}$ is thought of as being indexed by $\mu, \nu \in \mathbb{F}_q$.

$$\begin{aligned} C_{\mu,\nu} = \;& (1-(y_0+\mu)^{q-1}(y_0+\nu)^0 - (y_0+\mu)^{q-2}(y_0+\nu)^1 \\ & \cdots - (y_0+\mu)^0(y_0+\nu)^{q-1}) \\ & = 1 - \textstyle\sum_{i=0}^{q-1}(y_0+\mu)^{q-1-i}(y_0+\nu)^i \\ & = 1 - \frac{(y_0+\mu)^q - (y_0+\nu)^q}{y_0+\mu-(y_0+\nu)} \\ & \quad 1-(\mu-\nu)^{q-1} = \begin{cases} 1 & \mu = \nu \\ 0 & \mu \neq \nu \end{cases} \end{aligned}$$

■

We note that we are entirely free to choose ”virtual information symbols” in matrix $\tilde{\mathbf{r}}$. Let a sequence of information symbols be given that are to be encoded systematically in a codeword of a Hermitian code. The trick in obtaining a systematic encoding procedure is to choose the information symbols in $\tilde{\mathbf{r}}$ so that the mapping with A^{-1} and A'^{-1} respectively, gives the primary information symbols that we really want to encode.

Before we derive a systematic encoding procedure for Hermitian codes, we treat a somewhat simpler case, which will elucidate the idea of systematic encoding. Let $\hat{C}$ be a code on a Hermitian curve defined as

$$\begin{aligned} \hat{C} = \;& \{c \in \mathbb{F}_{q^2}^{q^3} \; : \; S_{a,b}(c) = 0, \\ & a = 0,1,\ldots,\hat{a} < q^2-1, \;\; b = 0,1,\ldots,q-1\} \end{aligned}$$

The code $\hat{C}$ has dimension $(q^2 - \hat{a} - 1)q$. The following algorithm may be used for systematic encoding of code $\hat{C}$.

B. Algorithm 1

1) *Write the $(q^2-\hat{a}-1)q$ information symbols in an array d of size $q \times (q^2-\hat{a}-1)$*
2) *Compute $\hat{\mathbf{r}} = Ad^T$*
3) *Encode the ith row of $\hat{\mathbf{r}}$ independently in a systematic way into codewords of the code*

   $$EC((\epsilon^{0+i(q+1)}, \epsilon^{1+i(q+1)}, \ldots, \epsilon^{\hat{a}(i)+i(q+1)}), q^2)$$

   Denote the resulting $q \times q^2$ matrix with $\hat{\mathbf{r}}'$.
4) *Compute columns $c_i = A_i^{-1}\hat{\mathbf{r}}'_i$*

Algorithm 1 yields a systematic encoding procedure for the code $\hat{C}$ because c is a code matrix by Lemma 2 and the first $(q^2-\hat{a}-1)\times q$ symbols are the original information symbols. The first $(q^2-\hat{a}-1)$ columns of d determine the first $(q^2-\hat{a}-1)$ columns of $\hat{\mathbf{r}}$. It is the first $(q^2-\hat{a}-1)$ columns of $\hat{\mathbf{r}}$ that contain the virtual information symbols for the encoding of the cyclic codes.

The situation for Hermitian codes is complicated by the fact that the codes E_i have different rates. Thus at some instance of the algorithm we have to process the columns that are in one part determined by information symbols and the other part is determined by redundancy symbols generated by the systematic encoders of the codes E_i. For simplicity, we will restrict our attention to codes $C(m)$ of dimension k that is less than (q^3-g-q).

Let $\phi_i \; : \; \mathbb{F}_{q^2}^{q^2-\hat{a}(i)-1} \longrightarrow \mathbb{F}_{q^2}^{q^2}$ be a systematic encoder for a code E_i. The input sequence to the encoder ϕ_i are symbols from an array $\tilde{\mathbf{r}} = \tilde{\mathbf{r}}_{i,j}$ for $j = 0,1,\ldots,q^2-\hat{a}(i)-2$.

We want to construct an algorithm that takes as input an array d of size $q \times q^2$ with arbitrarily chosen symbols in positions $(a,b) : b = 0,1,\ldots,q-1; a = 0,1,\ldots,q^2-\hat{a}(b)-2$ and zero in the remaining positions and that produces as output a code-array $\mathbf{c}$. Let $\hat{b}(j)$ be defined as the number of information symbols in the jth column of $\mathbf{d}$. The columns of $\mathbf{d}$ thus have the form $\mathbf{d}_j = (\mathbf{d}_{0,j}, \mathbf{d}_{1,j}, \ldots, \mathbf{d}_{\hat{b}(j)-1}, 0, 0, \ldots, 0)$.

We give a systematic encoder procedure in the following algorithm. During the procedure we also construct an array $\tilde{\mathbf{r}}$. The ith row of $\tilde{\mathbf{r}}$ is a codeword in E_i. Thus the first $q^2 - \hat{a}(i) - 1$ positions in the ith row of $\tilde{\mathbf{r}}$ determine the ith row of $\tilde{\mathbf{r}}$ completely.

C. Algorithm 2

The algorithm is shown in (6).

Theorem 1: Algorithm 2 computes a code array c of the Hermitian code $C = (ev_D(mP_\infty))^\perp$.

Proof: The matrix $\tilde{\mathbf{r}}$ in the algorithm satisfies the conditions $\tilde{\mathbf{r}}_j = A_j \; c_j$ and the i-th row of $\tilde{\mathbf{r}}$ is a codeword in E_i. Thus c is a code-array by Lemma 2. ■

Algorithm 2 outlines the mathematical procedure to achieve systematic encoding of a Hermitian code. The real difficulty

Algorithm 2 (6)

```
Input: An $q \times q^2$ array $\mathbf{d}$. An empty $q \times q^2$ array $\tilde{\mathbf{r}}$.
Iterations: For $j = 0, 1, \ldots, q^2 - 1$
    1) Compute the known part of $\tilde{\mathbf{r}}_j$ for $i = 0, 1, \ldots, q - 1 - \hat{b}(j)$ as
           $\tilde{\mathbf{r}}_{i,j} = (\phi_i((\tilde{\mathbf{r}}_{i,0}, \tilde{\mathbf{r}}_{i,1}, \ldots, \tilde{\mathbf{r}}_{i,q^2-\hat{a}(i)-2})))_j$
    2) Solve the equation
           $A_j(\mathbf{d}_{0,j}, \mathbf{d}_{1,j}, \ldots, \mathbf{d}_{\hat{b}(j)-1,j}, y_{\hat{b}(j),j}, y_{\hat{b}(j)+1,j}, \ldots, y_{q-1,j})^T = (\tilde{\mathbf{r}}_{0,j}, \tilde{\mathbf{r}}_{1,j}, \ldots, \tilde{\mathbf{r}}_{q-1-\hat{b}(j),j}, u_{q-\hat{b}(j),j}, u_{q-\hat{b}(j)-1,j}, \ldots, u_{q-1})^T$
       for $y_{\hat{b}(j),j}, y_{\hat{b}(j)+1,j}, \ldots, y_{q-1,j}, u_{q-\hat{b}(j),j}, u_{q-\hat{b}(j)-1,j}, \ldots, u_{q-1}$
    3) Set
           $c_i = (\mathbf{d}_{0,j}, \mathbf{d}_{1,j}, \ldots, \mathbf{d}_{\hat{b}(j)-1,j}, y_{\hat{b}(j),j}, y_{\hat{b}(j)+1,j}, \ldots, y_{q-1,j})$
           $\tilde{\mathbf{r}}_i = (\tilde{\mathbf{r}}_{0,j}, \tilde{\mathbf{r}}_{1,j}, \ldots, \tilde{\mathbf{r}}_{q-1-\hat{b}(j),j}, u_{q-\hat{b}(j),j}, u_{q-\hat{b}(j)-1,j}, \ldots, u_{q-1})$
```

lies in an efficient implementation of the algorithm. We give such an implementation in Section IV but before proceeding we will need a simple lemma.

Let A be any $n \times n$ matrix with inverse A^{-1}. We assume that the submatrix of A indexed by elements $i, j, i = n - l, \ldots, n - 1$ *and* $j = 0, 1, \ldots, l - 1$ is nonsingular. This will always be true for the cases that we are interested in by Lemma 1. Let I_l denote the $l \times l$ matrix and let $D(l)$ be a $n \times n$ matrix of the following form:

$$\mathbf{D(l)} = \left(\begin{array}{c|c} I_l & 0 \\ \hline P & 0 \end{array}\right)$$

such that

$$\mathbf{A^{-1}D} = \left(\begin{array}{c|c} 0 & 0 \\ \hline \tilde{P} & 0 \end{array}\right)$$

for a $l \times l$ matrix $\tilde{P}$. We note that $D(l)^T$ is just a systematic encoding matrix for a code which has a parity check matrix the first l rows of A^{-1}.

D. Lemma 5

Let $x_0, x_1, \ldots, x_{n-l-1}$ and $v_0, v_1, \ldots, v_{l-1}$ be given. The solution for $y_{n-l}, y_{n-l+1}, \ldots, y_{n-1}$ and $u_l, u_{l+1}, \ldots, u_{n-1}$ to the linear system of equations

$$A(x_0, x_1, \ldots, x_{n-l-1}, y_{n-l}, y_{n-l+1}, \ldots, y_{n-1})^T = (v_0, v_1, \ldots, v_{l-1}, u_l, u_{l+1}, \ldots, u_{n-1})^T$$

can be found with the following algorithm.

E. Algorithm 3

Algorithm 3 is given as shown in (7).

Proof: $A^{-1}\tilde{b}^T = A^{-1}D\hat{b}^T$ which proves that $y_0, y_1, \ldots, y_{n-l-1}$ equal zero.

Now it follows that

$$\begin{aligned}
&A(x_0, x_1, \ldots, x_{n-l-1}, y_{n-l}, y_{n-l+1}, \ldots, y_{n-1})^T \\
&= \tilde{b}^T + b^T \\
&= A(A^{-1}D\hat{b}^T) + b^T \\
&= D\hat{b}^T + b^T \\
&= D(v_0, v_1, \ldots, v_{l-1}, 0, 0, \ldots, 0)^T \\
&\quad -D(b_0, b_1, \ldots, b_{l-1}, 0, 0, \ldots, 0)^T + b^T \\
&= (v_0, v_1, \ldots, v_{l-1}, u_l, u_{l+1}, \ldots, u_{n-1})^T
\end{aligned}$$

■

IV. Efficient Implementation of a Systematic Encoder

Inspecting Algorithm 2 and Lemma 4, we see that we need modules for multiplication of an array with matrix A, A^{-1}, systematic encoding of codes E_j, and a systematic encoding module for codes D_l defined as

$$D_l = \{d \in \mathbb{F}_{q^2}^q : \sum_{j=0}^{q-1} A_{i,j}^{-1} d_j = 0, i = 0, 1, \ldots, l - 1\}$$

Before describing the modules in detail we give a black box description and the overall description of the implementation.

A. Module A: Multiplication with Matrix A, A'

The module has as parallel input a vector d of length q and produces as serial output the numbers $(Ad^T)_i$, $i = 0, 1, \ldots, q - 1$ during the next q clock cycles.

B. Module B: Multiplication with Matrix A^{-1}, A'^{-1}

Module B has as serial input a vector d of length q. After q clock cycles the parallel output is a vector $A^{-1}d^T$.

C. Module C: Systematic Encoding of Codes E_i

The module has a serial input of $q^2 - \hat{a}(i) - 1$ symbols and produce one symbol per clock cycle. The clocking frequency is $1/q$ of the overall clock rate.

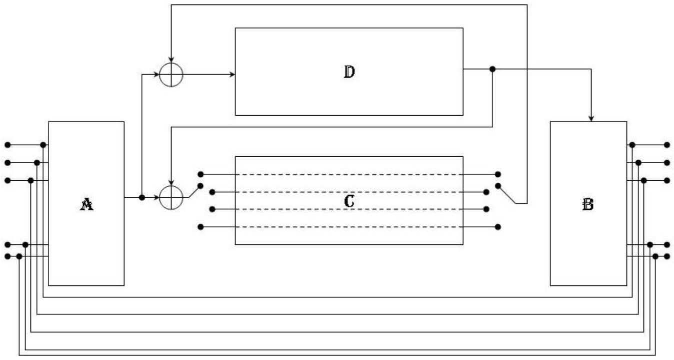

Fig. 1

OVERALL OUTLINE OF THE ENCODER CIRCUIT. SWITCHES a & b ARE SYNCHRONIZED AND ROTATE EVERY CLOCK CYCLE. THE CIRCUIT IS DESCRIBED IN THE TEXT.

D. Module D: Systematic Encoding of Codes D_l

Module D takes a serial input of length l and produces as serial output a codeword of length D_l.

E. Encoder

Figure 1 outlines the overall implementation. When the left hand input becomes valid, the output of Module A is added to the negative output of Module C, effectively implementing steps 1 and 2 of Algorithm 3. The sum is fed to Module D which implements step 3 of Algorithm 3. The output of Module D is combined with the output of Module A to implement step 4 of Algorithm 3. Simultaneously it is fed to module B of the implementation. After q clock cycles the output of Module B is added to the input thus implementing step 5 of Algorithm 3.

Module C can be implemented as an obvious modification of a systematic encoding circuits for RS codes [10].

Module D implements systematic encoding of a code with parity check matrix given by first l rows of matrix A^{-1}. From the form of matrices A^{-1}, we see that code D_l may be defined as

$$\begin{aligned} D_l = \quad & \{d \in \mathbb{F}_{q^2}^q : \textstyle\sum_{j=0}^{q-1} d_{q-1-j}(x_i)^j = d_0, x_0 = y_0, \\ & x_{s+1} = (y_0 + \gamma^s), s = 0, 1, \ldots, l-2\} \end{aligned}$$

and we can use standard encoding techniques for shortened cyclic codes which are modified in the obvious way.

V. FINAL REMARKS

A low complexity algorithm for systematic encoding and syndrome computation of Hermitian codes has been presented. The algorithm has a run time complexity of $O(n^{2/3})$ and is suitable for VLSI implementation. We give an outline for the encoder architecture, which uses as main blocks, q varying-rate Reed Solomon encoders. The architecture achieves a much lower space complexity in terms of finite field multipliers and memory elements when compared to earlier reported works.

VI. ACKNOWLEDGEMENT

The authors would like to thank Science Foundation Ireland, Claude Shannon Institute and Deutsche Forschungsgemeinschaf for supporting parts of this research work.

REFERENCES

[1] V. D. Goppa, *Codes on Algebraic Curves*, Soviet math. Dokl., 1981, 24, pp. 75-91

[2] B. E. Wahlen, and J. Jimenez, Performance Comparison of Hermitian and Reed-Solomon Codes, *Proc. MILCOM*, 1997.

[3] J. H. van Lint and T. A. Springer, Generalized Reed-Solomon Codes from Algebraic Geometry, *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 305-309, May 1987

[4] T. Yaghoobian and I. F. Blake, Hermitian Codes as Generalized Reed-Solomon Codes, *Design, Codes, Cryptography*, vol. 2, pp. 5-17, 1992

[5] B. Z. Shen, On Encoding and Decoding of the codes from Hermitian Curves, *Proc., Cryptography and Coding III*, vol. 45, M. Ganley, Ed., Oxford, UK, pp. 337-356, 1993

[6] J. Little, K. Saints and C. Heegard, On the structure of Hermitian Codes, *Journal of Pure and Applied Algebra*, vol. 121, pp. 293-314, 1997

$Algorithm\ 3$ (7)

1) $b^T = A(x_0, x_1, \ldots, x_{n-l-1}, 0, 0, \ldots, 0)^T$
2) $\hat{b}^T = (v_0, v_1, \ldots, v_{l-1}, 0, 0, \ldots, 0)^T - (b_0, b_1, \ldots, b_{l-1}, 0, 0, \ldots, 0)^T$
3) $\tilde{b}^T = D\hat{b}^T$
4) $(x_0, x_1, \ldots, x_{n-l-1}, y_{n-l}, y_{n-l+1}, \ldots, y_{n-1})^T = (x_0, x_1, \ldots, x_{n-l-1}, 0, 0, \ldots, 0)^T + A^{-1}\tilde{b}$
5) $(v_0, v_1, \ldots, v_{l-1}, u_l, u_{l+1}, \ldots, u_{n-1})^T = \tilde{b}^T + b^T$

[7] C. Heegard, J. Little and K. Saints, “Systematic Encoding via Gröbner bases for a class of Algebraic-Geometric Codes,” *IEEE Trans. Info. Theory,* vol. IT-41, pp. 1752-1762, Nov. 1995

[8] J. Chen and C. Lu, ”A Serial-In-Serial-Out Hardware Architecture for Systematic Encoding of Hermitian Codes via Gröbner Bases”, *IEEE Trans. Info. Theory*, Vol. 52, No. 8, pp. 1322-1332, August 2004

[9] H. Stichtenoth, *Algebraic Function Fields and Codes*, Springer-Verlag, Berlin, Germany 1993.

[10] G. Fettewis and M. Hassner, “A Combined Reed-Solomon Encoder and Syndrome Generator with Small Hardware Complexity” *Proc. Int. Symp. Circuits and Systems,* pp. 1871-1874, 1992.